DEVELOPMENT OF A FINITE STATE MACHINE FOR A SMALL UNMANNED AIRCRAFT SYSTEM USING EXPERIMENTAL DESIGN

THESIS
MARCH 2015

Jonathan D. Young, Major, USA

AFIT-ENS-MS-15-M-146

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENS-MS-15-M-146

DEVELOPMENT OF A FINITE STATE MACHINE FOR A SMALL UNMANNED AIRCRAFT SYSTEM USING EXPERIMENTAL DESIGN

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Jonathan D. Young, BA
Major, USA

March 2015

AFIT-ENS-MS-15-M-146

DEVELOPMENT OF A FINITE STATE MACHINE FOR A SMALL UNMANNED AIRCRAFT SYSTEM USING EXPERIMENTAL DESIGN

Jonathan D. Young, BA
Major, USA

Committee Membership:

Maj Brian B. Stone, PhD
Chair

Dr. Raymond R. Hill
Member

Dr. David R. Jacques
Member

Abstract

This research presents a methodology for improving the capability of a small unmanned aircraft system (SUAS) to autonomously track a moving ground vehicle. One drawback of the most common open source SUAS autopilot software, APM:Plane, is the inability to maintain a consistent following distance from the target vehicle under varying conditions defined by wind direction, wind speed, and target vehicle maneuver. Finite state machine (FSM) logic was developed to improve the APM:Plane software by reducing the variability in the following distance between the SUAS and the target vehicle. The FSM consists of 36 individual states defined by a combination of four wind directions, three wind speeds, and three ground maneuvers. Once the SUAS enters a particular state, the FSM modifies the default APM:Plane firmware parameter settings to optimal settings. The parameter settings for each state were determined from the statistical analysis of a sequence of designed experiments conducted in a simulated environment. During a real-world software validation experiment, the FSM reduced following distance variance by an average of 50 percent when compared to the default software settings.

Table of Contents

Page

List of Figures

Page

List of Tables

Page

List of Equations

DEVELOPMENT OF A FINITE STATE MACHINE FOR A SMALL UNMANNED AIRCRAFT SYSTEM USING EXPERIMENTAL DESIGN

I. Introduction

1.1 Context

Emerging technologies that increase the autonomous capability of unmanned systems are under development. Of particular interest to this research effort are autonomous enhancements to Small Unmanned Aircraft Systems (SUAS). When compared to their larger counterparts, SUAS can be used for many of the same applications and enjoy the advantage of being relatively inexpensive. Reduced cost and valuable capability point to increased future usage of SUAS.

One capability, autonomous vehicle tracking, is the focus of this research. This has applications in the United States Department of Defense that include convoy over-watch and surveillance. An SUAS able to provide valuable mission support autonomously would likely be fielded in large numbers. The ability of the end user to operate the SUAS without significant training and/or highly trained support personnel is one factor that would make this so. The significance of an SUAS capable of autonomous mission support cannot be overstated. Every convoy commander could enjoy a bird's eye view of the vehicles in his or her convoy and monitor the route for current threats. Every tactical unit could track a tagged vehicle to a target destination and obtain real-time information. In sum, adding this capability to lower echelon units is potentially life-saving.

1.2 Problem Background and Statement

The APM (formerly ArduPilotMega) autopilot running the APM:Plane firmware provides inexpensive autonomous capability to fixed-wing SUAS. The APM:Plane firmware is free, open source software written in C++ that receives frequent updates from a team of core developers.

The APM autopilot has a suite of sensors, including plug-in sensors, that can detect aircraft position, attitude (yaw, pitch, and roll), altitude, and speed. The autopilot can also measure environmental factors like air temperature, wind speed, and wind direction. The data produced by the autopilot is logged and available in real-time through the telemetry feed. APM:Plane was written to control the aircraft using the sensor information and react to environmental factors like wind. APM:Plane includes a number of input parameters that allow the user to adjust the autopilot for various airframes and change aircraft performance.

A concern is that the default APM:Plane parameter configuration is not optimized for maneuvers under various wind conditions. Thus, this research is focused on improving aircraft performance when maneuvering in the presence of wind. The maneuvers are those conducted in the process of following a ground vehicle.

Each combination of wind speed, wind direction, and ground maneuver can be considered a unique condition or state. There is likely a set of optimum software parameters unique to each state that improves aircraft performance. Because the APM:Plane parameters can be updated in real-time, as the SUAS transitions to a particular state, the appropriate parameters setting can be instantaneously uploaded.

Experimental design is well-suited to help discover optimum software parameters. Part of the experimental design process is developing experiments that are efficient and cost-effective. Hence, the process of finding improved parameters and enhancing the autonomous capability of the aircraft must use efficient experiments.

The primary concern of an end-user of an SUAS is likely reliable functionality. Any modifications should not negatively impact the reliability of the SUAS. Therefore, it is necessary to test autonomy enhancements.

1.3 Research Objective

This investigation into improving SUAS performance focuses on enhancing the aircraft's ability to maintain a consistent distance from the ground vehicle in the presence of wind and while the vehicle is performing a basic maneuver. That is, reducing the variability in following distance, defined as the horizontal distance from the ground vehicle to the aircraft, is the research objective. During the experimentation process altitude is fixed, consequently, horizontal distance is an appropriate substitute for straight-line distance from the ground vehicle to the aircraft.

1.4 Investigative Questions

In order to meet the aforementioned objective, this research focuses on answering the following questions:

1. Which APM:Plane control parameters — throttle slew rate, maximum bank angle, roll time constant, waypoint radius, waypoint loiter radius, and target airspeed — significantly impact variability in following distance?

2. Does changing parameters using state-based logic, where states are determined by wind direction, wind speed, and ground vehicle maneuver, reduce variability in following distance?
3. What set of parameter values specific to each state reduces variability in following distance?

1.5 Research Scope

There is inherent combinatorial complexity in the possible scenarios composed of sequences of ground vehicle maneuvers and varying wind conditions. To manage the complexity this research uses a number of rules. First, a typical ground vehicle route is decomposed into simple maneuvers. A ground vehicle generally conducts three maneuvers over the course of a typical route. The maneuvers are drive straight, execute a 90 degree turn, and execute a 180 degree turn or u-turn. A right 90 degree turn and a left 90 degree turn are equivalent. To facilitate replication further rules are necessary. The duration of each maneuver is 60 seconds. The vehicle travels at a ground speed of 25 miles per hour, only reducing speed to execute a turn. Once the turn is complete, the vehicle returns to 25 miles per hour. For the maneuvers that include a turn, the vehicle drives straight for ten seconds, executes the turn, and drives straight until 60 seconds have passed.

Second, like a vehicle route, wind direction has infinite possibilities. In order to discretize a continuous factor space, wind direction is relative to the aircraft heading and originates from one of three directions. The directions are headwind, tailwind, and crosswind. A headwind originates from -45 degrees to the 45 degrees. A crosswind

originates from 45 degrees to 135 degrees or -45 degrees to -135 degrees. A tailwind originates from -135 degrees to 135 degrees. See Figure 1 for an explanation of wind direction.

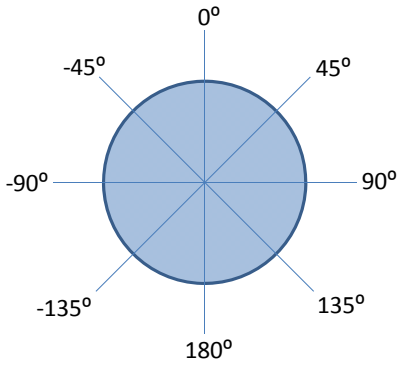


Figure 1: Wind Direction

Third, a crosswind originating from the left side of the aircraft is considered equivalent to a crosswind from the right for the straight and u-turn maneuvers. This simplification is not justifiable for the turn maneuver. Therefore, both left and right crosswinds are considered when estimating autopilot parameters for the turn maneuver.

Fourth, wind speed does not exceed the capability of the aircraft. The factor space for wind speed is set accordingly. This roughly equates to maximum wind speed being 50 percent or less of the aircraft cruise airspeed.

Because the experiments involve environmental factors, experiments must take place in a simulated environment where weather can be controlled. This also has the benefit of being more cost-effective and time-saving when compared to experimenting in a live environment. Moreover, given the limited time allotted for this research, extensive live environment experimentation cannot occur.

1.6 Methodology Overview

In order to reduce the variance of following distance, optimum autopilot parameters settings are estimated for each combination of wind speed, wind direction, and ground maneuver using statistical models. The estimated optimum parameters are organized into states. The states serve as the basis for a finite state machine (FSM) that is implemented using Python language scripts. The scripts allow the SUAS to select optimal parameter settings based on the prevailing wind conditions its sensors detect.

Optimum parameters settings are estimated using designed experiments. Designed experiments are also used to provide information on the FSM's efficacy and reliability. The majority of experiments occur in a simulated environment. As such, this research endeavors to verify that the simulator is an adequate substitute for conducting live experiments.

1.7 Thesis Preview

The remainder of this thesis is organized as follows. Chapter 2 briefly examines past Air Force Institute of Technology (AFIT) research related to this topic and the literature available on the autopilot control parameters that are at the core of this investigation. Chapter 3 lays out the methodology used to answer the investigative questions. Chapter 4 reports the results of the experiments described in Chapter 3 and provides answers to the investigative questions. Chapter 5 concludes the discussion of this research effort and provides recommendations for future work.

II. Literature Review

2.1 Chapter Overview

This chapter describes selected principles of experimental design and provides a brief definition of FSMs. It also examines the literature available on APM autopilot control parameters that are seemingly relevant to this investigation. Lastly, this chapter reviews past AFIT research efforts related to this research topic.

2.2 Experimental Design

Experimentation is the process of analyzing cause-and-effect relationships in a system by deliberately changing system inputs to produce a change in the system output. The inputs are termed factors and the output is known as the response. The experimentation process follows statistical principles that result in valid conclusions. The three basic statistical principles are randomization, replication, and blocking. Randomization means the individual runs of an experiment occur in random order. Replication refers to repeating independent runs to improve the estimation of error. Blocking is a technique used to exclude extraneous or nuisance factors and obtain a more precise estimate. [1]

Montgomery (2012) proposes guidelines for designing experiments. First, state the problem. Second, select the response. Third, select factors and factor levels. Factor levels refer to selecting the high and low settings of the system input used during the experiment. Fourth, choose an experimental design. Fifth, perform the experiment. Sixth, analyze the results using statistical methods. Last, draw conclusions and make recommendations. [1]

Choice of experimental design is a key element of experimenting efficiently. A design should be selected based on the expected empirical model. If the empirical model is expected to include first order terms, interaction terms, or higher order terms, the design must allow for their estimation. An experimental design used in this research effort is the Plackett-Burman. Plackett-Burman designs are suitable for experiments that require a number of runs that is a multiple of four. Of interest are usually experiments that require 12, 20, 24, 28, or 36 runs. For example, the 12-run Plackett-Burman design is a resolution III design that can estimate main effects for a system with eight factors. As it is a more efficient use of resources, the 12-run Plackett-Burman design may be preferred over a more popular fractional factorial design that would require 16 runs to estimate main effects for eight factors. [1]

Another design is the no-confounding design (NCD). The NCD is non-regular. It is first-order orthogonal and can estimate main effects. It can also provide estimates of all two-factor interactions where there is no complete confounding of any single two-factor interaction. This is an advantage over regular designs. For example, a 24-run NCD [2] can estimate main effects and all two-factor interactions for a system with six factors. A fractional factorial resolution IV design for six factors requires 16 runs. Unfortunately, many of the two-factor interactions are aliased with one another. If two-factor interactions are expected to be significant, but the expected number of significant two-factor interactions is unknown, a resolution IV design would require additional experimentation to provide individual estimates for two-factor interactions that are aliased. The fractional factorial design for six factors that can estimate all two-factor

interactions without aliasing requires 32 runs. Therefore, in this scenario the 24-run NCD holds the advantage and is more efficient. [1]

The definitive screening design (DSD) is a three-level design that estimates main effects, two-factor interactions, and quadratic effects. According to Jones (2011), "definitive screening designs offer the following advantages:

- For continuous factors, the number of required runs is only one more than twice the number of factors. Categorical factors require two more than twice the number of factors.
- Unlike resolution III designs, main effects are completely independent of two-factor interactions. As a result, estimates of main effects are not biased by the presence of active two-factor interactions, regardless of whether the interactions are included in the model.
- Unlike resolution IV designs, two-factor interactions are not completely confounded with other two-factor interactions, although they might be correlated.
- Unlike resolution III, IV, and V designs with added center points, all quadratic effects are estimable in models comprised of any number of linear and quadratic main-effects terms.
- Quadratic effects are orthogonal to main effects and not completely confounded (though correlated) with interaction effects.
- With six through (at least) twelve factors, the designs are capable of estimating all possible full quadratic models involving three or fewer factors with very high levels of statistical efficiency." [3]

Traditionally, the estimation of quadratic effects is done using a central composite design (CCD). For six factors, a CCD requires 46 runs, which includes two center point runs. This design can estimate all main effects, two-factor interactions, and quadratic effects. However, a CCD is usually part of a sequential experimentation process. Normally prior screening experiments would have reduced the number of factors and the CCD would be much more manageable. If it is known beforehand that estimations of all the main, two-factor interaction, and quadratic effects are not required, a DSD can screen and estimate effects, even quadratic effects, in one experiment. A DSD for six factors requires 13 runs. Additional runs can be added by creating a matrix for eight factors and deleting the last two columns. Using this method for six factors, the DSD would have 17 runs and is capable of estimating more effects without aliasing. [3]

2.3 Finite State Machines

An FSM is a model of computation. It is a useful tool to model a system where inputs cause the system to transition to a particular condition or state. The state depends on the value of the inputs. An FSM has the following characteristics:

- "The system must be describable by a finite set of states.
- The system must have a finite set of inputs and/or events that can trigger transitions between states.
- The behavior of the system at a given point in time depends upon the current state and the input or event that occur at that time.
- For each state the system may be in, behavior is defined for each possible input or event.

- The system has a particular initial state." [4]

A simple example of an FSM is a turnstile (see Figure 2) one might find at a train station. The turnstile has two states, locked and unlocked. Inserting a coin in the turnstile causes the transition from locked to unlocked. Once the individual who inserted the coin passes through the turnstile, it transitions back to locked.

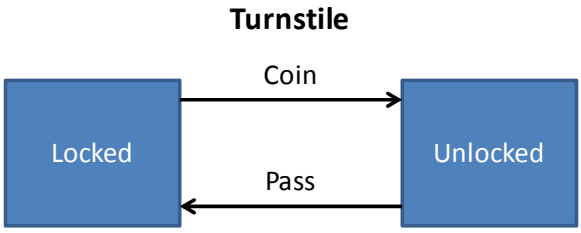


Figure 2: Simple Finite State Machine Example

2.4 APM:Plane Firmware Parameters

The APM:Plane firmware has more than 300 configurable parameters. The definitions of a few parameters that are relevant to this research effort are covered. Parameters of interest are those that impact aircraft responsiveness and navigation. Aircraft responsiveness refers to those attitude parameters that affect two-dimensional movement. Not under consideration are those parameters that affect the aircraft's ability to change altitude. Additionally, navigation parameters that impact waypoint characteristics are considered. Parameter definitions follow:

1. Throttle slew rate: "Maximum percentage change in throttle per second. A setting of 10 means the throttle will not change by more than 10% of the full throttle range in one second." [5]
    a. Range: 0 to 100

b. Increment: 1

c. Units: percent

d. Default: 100

2. Waypoint radius: "Defines the maximum distance from a waypoint that, when crossed, indicates the waypoint is complete. To prevent the aircraft from looping around the waypoint until it achieves the set waypoint radius, an additional check is made to see if the aircraft has crossed a "finish line". The finish line is a line that passes through the waypoint and is perpendicular to the flight path from the previous waypoint. If that finish line is crossed, then the waypoint is considered complete." [5]

    a. Range: 1 to 32767

    b. Increment: 1

    c. Units: meters

    d. Default: 90

3. Waypoint loiter radius: "Defines the distance from the waypoint center the plane will maintain during a loiter. If you set this value to a negative number then the default loiter direction will be counter-clockwise instead of clockwise." [5]

    a. Range: -32767 to 32767

    b. Increment: 1

    c. Units: meters

    d. Default: 60

4. Maximum bank angle: "The maximum commanded bank angle in either direction." [5]
    a. Range: 0 to 9000
    b. Increment: 1
    c. Units: centidegrees
    d. Default: 4500
5. Target airspeed: "Airspeed in cm/s to aim for when airspeed is enabled in auto mode. This is a calibrated (apparent) airspeed." [5]
    a. Units: cm/s
    b. Default: 1200
6. Roll time constant: "This controls the time constant in seconds from demanded to achieved bank angle. A value of 0.5 is a good default and will work with nearly all models. Advanced users may want to reduce this time to obtain a faster response but there is no point setting a time less than the aircraft can achieve." [5]
    a. Range: 0.4 to 1.0
    b. Increment: 0.1
    c. Units: seconds
    d. Default: 0.5

These parameter definitions are helpful in determining the precise effect each has on the aircraft and whether or not each parameter is a suitable candidate as a factor for experimentation. Additionally, the definitions are helpful in determining initial factor settings.

2.5 Past AFIT Research

Livermore (2014) proposed a method to calculate an optimal flight path for a small unmanned aircraft following a moving ground vehicle. To accomplish this, Livermore developed a cost function, which minimized aircraft control effort and deviation from a desired ground vehicle following distance, as the basis to calculate an optimal flight path. Control effort was represented by the aircraft roll rate. Following distance was termed "slant range". Slant range is the straight-line distance between the aircraft and the ground vehicle. Where *X* represents latitude, *Y* represents longitude, and *h* represents relative altitude, the formula to calculate slant range, *SR*, is given in Equation 1.

$$SR = \sqrt{\left(X_{gv} - X_{uav}\right)^2 + \left(Y_{gv} - Y_{uav}\right)^2 + h^2}$$

Equation 1: Slant Range

Where α is the weight factor and *u* is the aircraft roll rate, the cost function, *J*, is given in Equation 2.

$$J = \int_{t_0}^{t_f} \left[\alpha \left(\frac{SR - SR_{desired}}{SR_{desired}} \right)^2 + (1-\alpha) \left(\frac{u}{u_{\max}} \right)^2 \right] dt$$

Equation 2: Cost Function

To generate the flight path, Livermore employed a Matlab® function, fmincon, which uses the interior point method to evaluate the cost function. This solution provided a basis by which Livermore evaluated less computationally intensive heuristic methods of calculating an optimal flight path. Given the hardware limitations of the APM autopilot,

the Matlab® function could not be implemented. As one of the objectives was to develop a solution that could be implemented in real time and onboard the aircraft, a more feasible heuristic-based approach was used. [6]

Neal (2014) implemented the heuristic-based approach by modifying the autopilot firmware to calculate *J* and *SR* over time and providing that data as part of the aircraft telemetry. Equation 3 is Neal's simplified cost function.

$$J_t = \alpha\left(\frac{SR_t - SR_{desired}}{SR_{desired}}\right)^2 + (1-\alpha)\left(\frac{u_t}{u_{\max}}\right)^2$$

Equation 3: Simplified Cost Function

Neal designed an experiment to find autopilot parameter settings that minimize the response, *J*. The parameters Neal chose as factors were loiter radius, loiter range, and lead time. Neal found loiter radius to be the significant factor in minimizing *J*. Neal developed an FSM which monitored the reported values of *J* and *SR* to determine presence in one of three states, standard target tracking, low range target tracking, or high range target tracking. The FSM was designed to keep the aircraft in the standard target tracking state. If *J* or *SR* falls outside established thresholds, the autopilot increases or decreases the loiter radius. An increase occurs if the aircraft is in the low range tracking state. Conversely, a decrease occurs if the aircraft is in the high range tracking state. [7]

The joint effort between Neal and Livermore resulted in a flight path with a lower cost function value when compared to the flight path generated by the aircraft using default autopilot firmware. The default path cost function value was 113 times greater than the Matlab® generated or optimal flight path. Boasting an improvement, the

heuristic approach reduced this ratio to only 7.5 times greater than the optimal flight path. [6]

An earlier effort by Lozano (2011) employed regression analysis and experimental design in an attempt to improve SUAS endurance. The unmanned aircraft under investigation was an electric Overhead Watch and Loiter (OWL) with an onboard Procerus Technologies® Kestrel autopilot. The OWL was derived from the Raven RQ-11B. Fuel efficiency in terms of battery power conservation represented endurance. [8]

Lozano used regression analysis to determine that throttle percentage was the factor that most significantly affected battery power as represented by amperage. Lozano developed a predictive regression model with amperage as the response and throttle servo percentage as the independent variable. [8]

Realizing that throttle percentage could not be directly controlled on the Kestrel autopilot, Lozano used a designed experiment to research other factors that affect throttle percentage and, thus, battery amperage. Lozano selected throttle<airspeed Kp, pitch<altitude Kp, cruise airspeed, and throttle time constant as the Kestrel autopilot parameters to test. Lozano developed models for both low and high wind conditions. Cruise airspeed was the common significant factor to both models. Using the optimum airspeed settings resulted in improving aircraft endurance by approximately 30 percent under both low and high wind conditions. [8]

III. Methodology

3.1 Chapter Overview

This chapter addresses the methods used to answer the research questions, namely:

1. Which APM:Plane control parameters — throttle slew rate, maximum bank angle, roll time constant, waypoint radius, waypoint loiter radius, and target airspeed — significantly impact variability in following distance?
2. Does the implementation of state-based logic, where states are determined by wind direction, wind speed, and ground vehicle maneuver, reduce variability in following distance?
3. What set of parameter values specific to each state reduces variability in following distance?

The answers to these questions are used to craft an FSM implemented through Python language scripting that allows the aircraft to dynamically select improved parameters settings based upon the prevailing state.

3.2 Equipment and Materials

Experimentation involves an SUAS operating in a simulated and live environment where the aircraft follows a ground vehicle. Generally, an SUAS consists of an air vehicle, a ground control station, communications hardware, and related support equipment and personnel. In the simulated environment, a Windows 7 computer with the necessary software and hardware serves to represent both the SUAS and the ground vehicle.

For the simulated environment, the following hardware and software are used: a simulated Sig Rascal 110 modeled in FlightGear 3.0.0, an open source flight simulator; an APM 2.6 autopilot loaded with APM:Plane 3.1.1, open source hardware-in-the-loop (HIL) firmware; and Mission Planner 1.3.17, open source ground control station software. Franson GpsGate Client 2.6 software is used to simulate the ground vehicle. The APM is connected to the computer via USB.

Mission Planner interfaces with the APM autopilot and allows control of the simulated Sig Rascal 110 in FlightGear. FlightGear provides the APM with simulated environment inputs such as location, weather, and terrain. GpsGate supplies a virtual GPS receiver that can play recorded GPS tracks and transmit the GPS data to Mission Planner. With the APM set to follow-me mode, the simulated rascal follows the GPS track. The GPS track represents the ground vehicle performing a maneuver. See Figure 3.

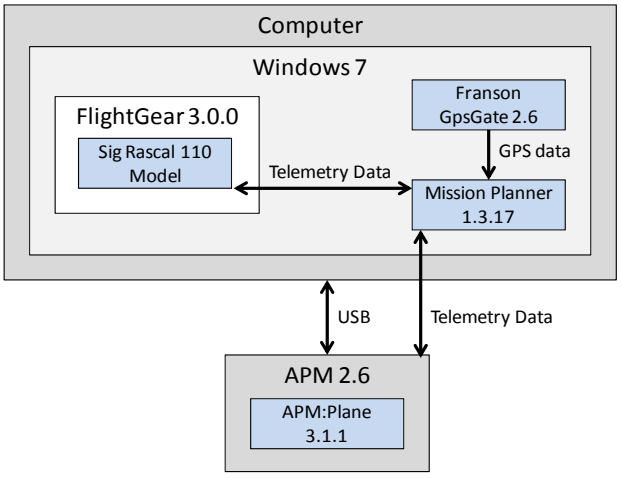


Figure 3: Architecture Diagram – Simulated Environment

In the live environment the following hardware and software are used: a Sig Rascal 110 with an APM 2.6 autopilot loaded with APM:Plane 3.1.1 flight firmware; a Windows 7 computer loaded with Mission Planner 1.3.17 and Franson GpsGate 2.6; and a safety pilot. A communications modem and GPS receiver are attached via USB to the computer. The communications modem allows Mission Planner to monitor and control the Sig Rascal 110. The GPS receiver provides GPS location data to Mission Planner which in turn directs the Sig Rascal 110 to follow. The computer is placed in a ground vehicle which performs maneuvers while the Sig Rascal 110 attempts to autonomously follow. See Figure 4.

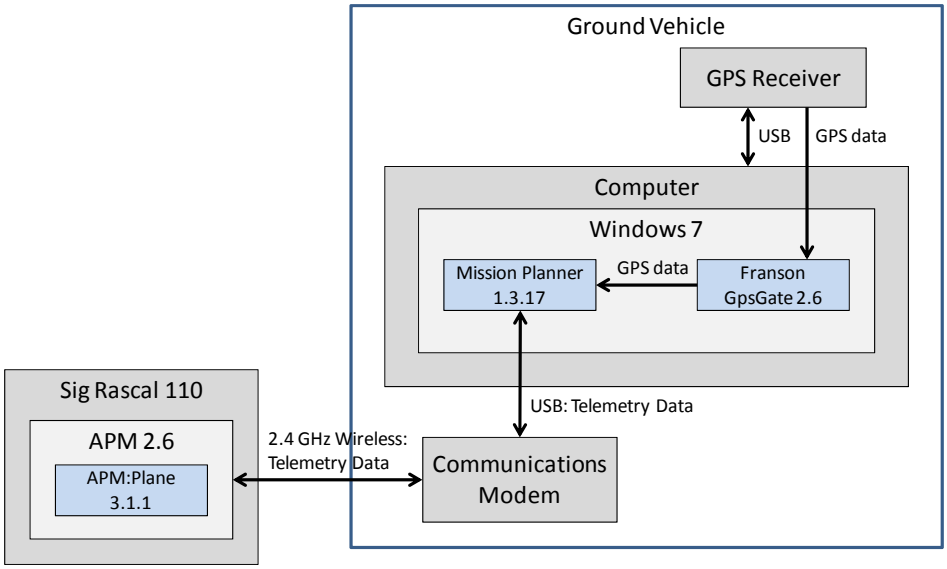


Figure 4: Architecture Diagram – Live Environment

3.3 Initial Conditions

Starting conditions are standardized for each run to ensure consistency of results. Before each run, wind direction and wind speed are configured to the appropriate settings using the FlightGear manual weather configuration tool found under the environment tab.

The APM:Plane control parameters are set to the appropriate levels. Next, the aircraft is put in a loiter above the ground vehicle at the starting point until the live telemetry feed shows an accurate reading of wind direction. This usually requires the aircraft to loiter one to two times. The ground vehicle begins its maneuver when the aircraft is directly behind the ground vehicle. At this point the run has begun and the start time is recorded. Each run lasts 45 seconds.

3.4 Factors

For this research effort, following distance variance is the response of interest. The parameters, environmental conditions, and other aspects that impact following distance variance are considered and from those the factors for investigation are selected. A fishbone diagram helps organize thoughts (see Figure 5 below). The definition of following distance variance is solely concerned with variability in two-dimensional movement. There is no attempt to impact aircraft altitude. The potential factors are divided into four categories: aircraft responsiveness, navigation settings, ground vehicle, and environment. Ground vehicle and environment are external to the SUAS and are the primary source of following distance variability. In order to react to ground vehicle behavior and changing environmental conditions, aircraft responsiveness and navigation settings are addressed.

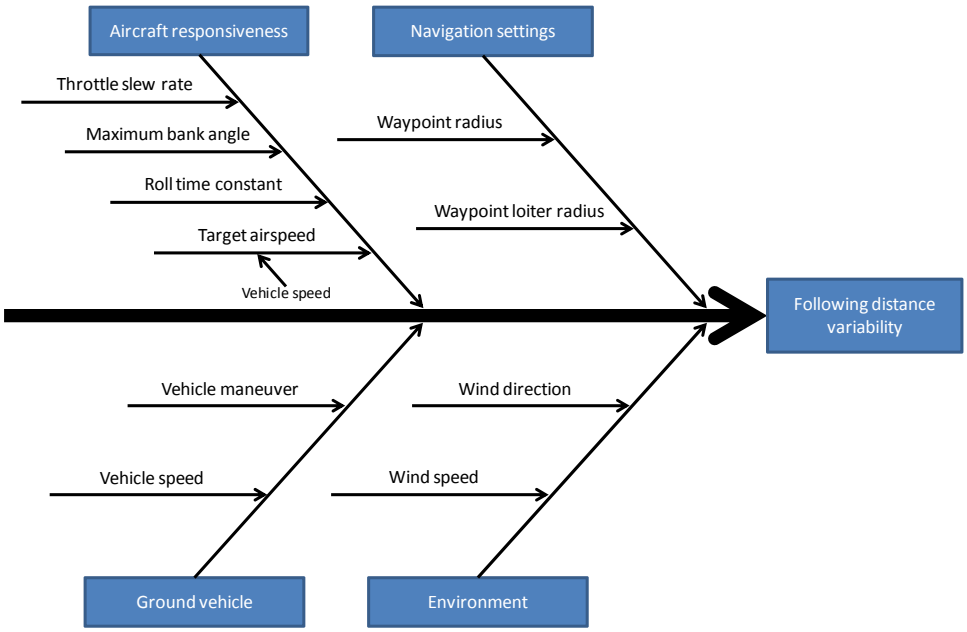


Figure 5: Fishbone Diagram

Improving aircraft responsiveness, ability to react to changes in ground vehicle direction and speed, as well as, changes in wind direction and speed, likely improves following distance consistency. The control parameters maximum bank angle and roll time constant impact aircraft turn agility, whereas, throttle slew rate and target airspeed impact the aircraft's ability to increase or reduce speed and/or distance from the ground vehicle. The APM autopilot treats the ground vehicle as a moving waypoint. Thus, navigation settings involving waypoints should also impact following distance variability.

After consideration, throttle slew rate, waypoint radius, waypoint loiter radius, maximum bank angle, target airspeed, roll time constant, wind speed, and wind direction are chosen as initial experimental factors. Vehicle maneuver is used as a categorical factor. Wind speed and wind direction are only viable factors for experiments conducted in the simulated environment. In later iterations, target airspeed is excluded as a factor

and set equal to ground vehicle speed. Wind direction is also changed to be a categorical factor. Each experiment is a combination of the wind direction and ground vehicle maneuver.

3.5 Factor Levels

The APM:Plane control parameter settings can impact the ability of the aircraft to remain in flight. An experiment is conducted to ensure factor levels are appropriately set to avoid crashing the aircraft during subsequent experimentation. Additionally, this experiment serves to verify procedures and proper functioning of software and hardware. The purpose of this experiment is not to screen factors.

Because experiments take place in the simulated environment, environmental factors of interest, namely wind direction and wind speed are controlled. Wind direction and wind speed are added as continuous factors. Initially, factor levels are set as wide as seemed feasible. Table 1 shows the factors and initial factor levels.

Table 1: Initial Factor Levels

Factor (Continuous)	Low	Center	High
Throttle slew rate	50%	75%	100%
Waypoint radius	40 meters	90 meters	140 meters
Waypoint loiter radius	30 meters	60 meters	90 meters
Max bank angle	20 degrees	50 degrees	80 degrees
Target airspeed	11.18 m/s	13.41 m/s	15.65 m/s
Roll time constant	0.4 second	0.7 second	1.0 second
Wind speed	3 knots	7 knots	11 knots
Wind direction (relative)	0 degrees	90 degrees	180 degrees
Factor (Categorical)			
Vehicle maneuver	Straight	Turn	U-turn

This experiment employs a 12-run Plackett-Burman design — 12 runs at the design space extremes and one center run. The experiment uses only one ground vehicle maneuver, the turn. The design matrix is in Table 2.

Table 2: Plackett-Burman Design Matrix

Run	Throttle slew rate	Waypoint radius	Waypoint loiter	Max bank angle	Target airspeed	Roll time constant	Wind speed	Wind direction
1	1	1	1	1	1	1	1	1
2	1	-1	-1	1	-1	1	1	1
3	1	-1	1	1	1	-1	-1	-1
4	0	0	0	0	0	0	0	0
5	-1	1	-1	1	1	1	-1	-1
6	1	1	1	-1	-1	-1	1	-1
7	1	-1	-1	-1	1	-1	-1	1
8	-1	-1	1	-1	-1	1	-1	1
9	-1	1	1	1	-1	-1	-1	1
10	-1	-1	1	-1	1	1	1	-1
11	1	1	-1	-1	-1	1	-1	-1
12	-1	-1	-1	1	-1	-1	1	-1
13	-1	1	-1	-1	1	-1	1	1

The 12-run Plackett-Burman experiment was instrumental in determining proper factor levels and verifying selected initial conditions for each experimental run to be used throughout the experimentation process. This experiment also provided encouraging evidence that relevant factors were selected as a number of them were found to be statistically significant. The JMP® output is found in Table 3. As mentioned previously, the purpose of this experiment was only to adjust factor levels, not screen factors.

Table 3: Plackett-Burman JMP® Output

Effect Tests

Source	Nparm	DF	Sum of Squares	F Ratio	Prob > F
Throttle slew rate	1	1	989277479	6.7680	0.0600
Waypoint radius	1	1	849220373	5.8098	0.0735
Waypoint loiter radius	1	1	18334779.6	0.1254	0.7411
Max bank angle	1	1	3969000300	27.1533	0.0065*
Target airspeed	1	1	1095663443	7.4958	0.0520
Roll time constant	1	1	102762011	0.7030	0.4489
Wind speed	1	1	204438740	1.3986	0.3024
Wind direction	1	1	2479542200	16.9634	0.0146*

During the course of the experiment, it became very apparent that the maximum bank angle low setting was too low. This was the most significant finding. At twenty degrees, the aircraft could scarcely execute a turn. Add to that a higher speed headwind and the aircraft was blown far off course. This resulted in overemphasizing maximum bank angle and requiring too much time to complete a run at the low setting. Setting maximum bank angle at 20 degrees did not allow the aircraft to achieve the waypoint loiter radius, nullifying any effect this factor may have had.

Following completion of the formal experiment, a few informal runs were conducted to further investigate a feasible low setting for maximum bank angle. It was observed that a maximum bank angle of 30 degrees appeared to be an acceptable low setting and allowed the aircraft to reasonably execute a turn.

It is standard practice to model variance using a log transformation of the response. The appropriateness of this transformation was investigated using the Box-Cox transformation technique and the log transform was selected. The power parameter, $\lambda=0$, corresponds to a log transformation, which can be seen in the interval depicted in Figure 6.

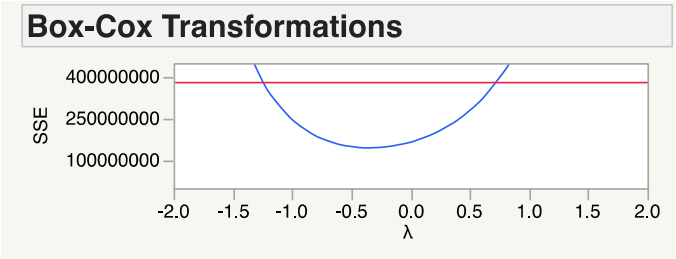


Figure 6: Plackett-Burman Box-Cox Transformation

3.6 Screening and Model Selection

With amended factor levels, the next step is to screen the factors, check ANOVA assumptions, examine the suitability of a transformation of the response, and check for pure quadratic curvature. Erring on the side of caution, a larger design to allow for estimation of main effects and two-factor interactions is selected. A 24-run NCD [2] augmented with three center runs ensures adequate ability to estimate desired effects, check for curvature, and preclude the need for further experimentation as long as lack of fit for the first order model is not detected. The number of continuous factors is reduced to six. Wind direction is now considered a categorical factor and target airspeed is set to match ground vehicle speed. There are nine unique combinations of vehicle maneuver and wind direction. If no curvature is detected, thus ruling out the need for quadratic terms in the model, the 24-run NCD is repeated for each of the nine combinations. The run order for each combination is randomized. The factor levels for the 24-run NCD are in Table 4.

Table 4: NCD Factor Levels

Factor (Continuous)	Low	Center	High
Throttle slew rate	50%	75%	100%
Waypoint radius	40 meters	90 meters	140 meters
Waypoint loiter radius	30 meters	60 meters	90 meters
Max bank angle	30 degrees	55 degrees	80 degrees
Roll time constant	0.4 second	0.7 second	1.0 second
Wind speed	3 knots	7 knots	11 knots
Factor (Categorical)			
Vehicle maneuver	Straight	Turn	U-turn
Wind direction (relative)	0 degrees	90 degrees	180 degrees

Table 5 is the NCD design matrix for a turn with a crosswind.

Table 5: NCD Design Matrix

Run	Throttle slew rate	Waypoint radius	Waypoint loiter	Max bank angle	Roll time constant	Wind speed
1	-1	-1	1	-1	-1	1
2	-1	-1	1	1	-1	1
3	-1	-1	-1	1	1	-1
4	0	0	0	0	0	0
5	-1	-1	1	1	1	-1
6	1	1	1	1	-1	-1
7	1	1	1	1	1	1
8	-1	1	1	-1	-1	-1
9	-1	-1	-1	-1	-1	-1
10	0	0	0	0	0	0
11	-1	1	-1	-1	1	1
12	-1	1	-1	1	-1	1
13	1	1	-1	-1	-1	1
14	1	1	1	-1	-1	1
15	-1	1	1	1	1	1
16	1	-1	1	-1	1	1
17	-1	1	-1	1	-1	-1
18	1	-1	1	1	-1	-1
19	-1	-1	-1	-1	1	1
20	1	1	-1	1	1	-1
21	-1	1	1	-1	1	-1
22	1	-1	-1	1	1	1
23	1	-1	1	-1	1	-1
24	1	-1	-1	1	-1	1
25	0	0	0	0	0	0
26	1	-1	-1	-1	-1	-1
27	1	1	-1	-1	1	-1

Once data are collected, the stepwise regression tool found in JMP® 11 is used to select the model for each combination. Primarily, the p-value threshold is used as the

stopping rule, checking the resulting model against the Akaike information criterion (AIC) and Mallows' Cp.

The 24-run NCD experiment, a ground vehicle turn with a crosswind, resulted in confirming the validity of a log transformation of the response and the presence of pure quadratic curvature (see Table 6 and Figure 7). Thus, from this point forward the natural logarithm of the following distance variance was used as the response and an experimental design capable of estimating quadratic effects was used.

Table 6: NCD JMP® Output with Curvature

Effect Tests

Source	Nparm	DF	Sum of Squares	F Ratio	Prob > F
Throttle slew rate	1	1	0.0060938	0.0519	0.8223
Waypoint radius	1	1	0.0023400	0.0199	0.8893
Waypoint loiter radius	1	1	1.2596525	10.7205	0.0040*
Max bank angle	1	1	9.8910947	84.1798	<.0001*
Roll time constant	1	1	0.0000967	0.0008	0.9774
Wind speed	1	1	0.0514447	0.4378	0.5161
Curvature	1	1	1.1881285	10.1118	0.0049*

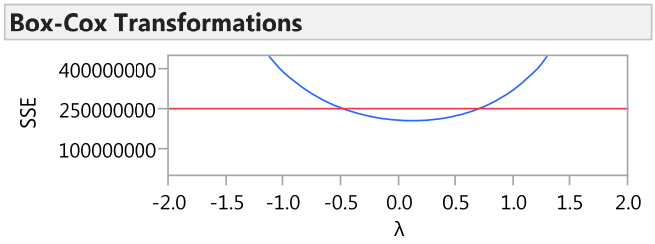


Figure 7: NCD Box-Cox Transformation

The NCD experiment also caused a reevaluation of the low setting for maximum bank angle. Observing the runs, thirty degrees still appeared to be too low and overly

hinder the aircraft's ability to execute turns. The low setting for future experimentation was set at forty degrees.

Because curvature was detected and quadratic terms are required in the models, a three-level, 17-run DSD that can estimate main effects, interactions, and quadratic terms is employed. An example DSD design matrix is found in Table 7.

Table 7: DSD Design Matrix

Run	Throttle slew rate	Waypoint radius	Waypoint loiter	Max bank angle	Roll time constant	Wind speed
1	1	0	1	1	-1	1
2	-1	-1	1	-1	1	1
3	-1	1	0	-1	-1	1
4	-1	-1	-1	1	-1	1
5	-1	-1	1	1	0	-1
6	0	1	1	1	1	1
7	1	1	-1	-1	0	1
8	0	-1	-1	-1	-1	-1
9	1	-1	0	1	1	-1
10	-1	1	-1	1	1	0
11	1	-1	1	-1	-1	0
12	1	-1	-1	0	1	1
13	0	0	0	0	0	0
14	1	1	-1	1	-1	-1
15	-1	1	1	0	-1	-1
16	1	1	1	-1	1	-1
17	-1	0	-1	-1	1	-1

3.7 Simulator Validation

Validation of the model results poses a challenge. Experimentation in a simulated environment allows control of wind speed and wind direction. Obviously, live experimentation does not provide the same convenience. Therefore, no attempt is made to validate models, but rather, validate the simulated environment as a satisfactory substitute for a live environment. For this purpose, a live experiment is conducted first. Then, replication of the environmental conditions in the simulated environment occurs and the experiment is repeated. A visual analysis is performed by comparing the route

trace of each live run with its simulated run counterpart. Finding the two route traces similar provides evidence that the results of experiments conducted in the simulated environment can reasonably approximate results of those same experiments performed under near identical conditions in a live environment.

Live environment experimentation can be fraught with unanticipated problems that result in delays or cancellations. An experiment executed under potentially problematic conditions must be designed to allow for collection of statistically meaningful results in the unfortunate event that the experiment is cut short. For this validation experiment, the JMP® 11 custom design tool is used to generate a test matrix that is organized into blocks of four runs each in order to reduce the impact of being unable to complete the experiment. Table 8 is the test matrix.

Table 8: HIL Validation Test Matrix

Run	Throttle slew rate	Waypoint radius	Waypoint loiter	Max bank angle	Roll time constant
1	-1	-1	1	-1	-1
2	1	1	1	1	-1
3	-1	-1	-1	-1	1
4	1	-1	1	1	1
5	1	1	-1	1	-1
6	1	-1	-1	1	-1
7	1	-1	1	-1	1
8	-1	1	1	-1	-1

3.8 Finite State Machine

With the models selected for each combination of the ground vehicle maneuver and wind direction, desirability functions found in the JMP® 11 Desirability Profiler find the settings that minimize the following distance variance for each combination of ground vehicle maneuver, wind direction, and wind speed. Each combination becomes a state in the FSM. The APM:Plane control parameters unique to each state are recorded in a

matrix. A matrix is developed for each ground maneuver. Under APM:Plane firmware 3.1.1, the aircraft cannot recognize ground vehicle maneuvers and, therefore, this aspect of the FSM is not tested. Under APM:Plane firmware 3.1.1, the aircraft can, however, provide information on wind direction and wind speed, which allows for testing. Mission Planner reports this information in the aircraft telemetry data.

Mission Planner possesses the capability to execute Python language scripts. Scripts are written for each ground maneuver that accept wind direction and wind speed as inputs and output APM:Plane control parameters settings optimized to reduce the variability in ground vehicle following distance. The script is executed under the same initial conditions described earlier.

3.9 Finite State Machine Validation

Validation of the models that serve as the basis of the FSM occurs by executing experiments in both the simulated and live environments that compare the following distance variance of a run using default parameters with the variance of a run using the experimentally suggested parameters. The variances are compared using the hypothesis test shown in Equation 4.

$$H_0 : \sigma_1^2 = \sigma_2^2$$
$$H_1 : \sigma_1^2 > \sigma_2^2$$

Equation 4: Hypothesis Test for the Equality of Two Variances

The ratio of the sample variances follows an F distribution. The test statistic is found in Equation 5.

$$F = \frac{s_1^2}{s_2^2}$$

Equation 5: F-test for the Equality of Two Variances

Where α=0.05 and n represents the sample size, the null hypothesis is rejected if the condition of Equation 6 is met.

$$F > F_{\alpha, n_1 - 1, n_2 - 1}$$

Equation 6: Rejection Region for the F-test of the Equality of Two Variances

If the larger variance belongs to the run executed under default parameter settings and the difference is significant, then there is sufficient evidence to show that the FSM reduces variability in ground vehicle following distance.

In the simulated environment, the FSM is tested for every ground vehicle maneuver conducted under a headwind, crosswind, and tailwind at wind speeds of 3 knots and 11 knots. This results in a 48-run, randomized test matrix where identically conditioned runs at default and improved parameters are conducted pairwise. The test matrix is in Table 9.

Table 9: Simulated Environment Test Matrix

Run	Wind Direction	Wind Speed	Maneuver	Parameters
1	Crosswind Left	11	Turn	Default
	Crosswind Left	11	Turn	Script
2	Headwind	3	Straight	Default
	Headwind	3	Straight	Script
3	Crosswind Right	11	U-turn	Default
	Crosswind Right	11	U-turn	Script
4	Crosswind Left	11	Straight	Default
	Crosswind Left	11	Straight	Script
5	Crosswind Left	11	U-turn	Default
	Crosswind Left	11	U-turn	Script
6	Headwind	11	Straight	Default
	Headwind	11	Straight	Script
7	Headwind	11	Turn	Default
	Headwind	11	Turn	Script
8	Crosswind Left	3	U-turn	Default
	Crosswind Left	3	U-turn	Script
9	Crosswind Left	3	Straight	Default
	Crosswind Left	3	Straight	Script
10	Crosswind Left	3	Turn	Default
	Crosswind Left	3	Turn	Script
11	Crosswind Right	3	Straight	Default
	Crosswind Right	3	Straight	Script
12	Headwind	11	U-turn	Default
	Headwind	11	U-turn	Script
13	Headwind	3	U-turn	Default
	Headwind	3	U-turn	Script
14	Crosswind Right	3	Turn	Default
	Crosswind Right	3	Turn	Script
15	Tailwind	11	U-turn	Default
	Tailwind	11	U-turn	Script
16	Tailwind	3	U-turn	Default
	Tailwind	3	U-turn	Script
17	Crosswind Right	3	U-turn	Default
	Crosswind Right	3	U-turn	Script
18	Tailwind	3	Turn	Default
	Tailwind	3	Turn	Script
19	Tailwind	3	Straight	Default
	Tailwind	3	Straight	Script
20	Crosswind Right	11	Turn	Default
	Crosswind Right	11	Turn	Script
21	Tailwind	11	Turn	Default
	Tailwind	11	Turn	Script
22	Crosswind Right	11	Straight	Default
	Crosswind Right	11	Straight	Script
23	Tailwind	11	Straight	Default
	Tailwind	11	Straight	Script
24	Headwind	3	Turn	Default
	Headwind	3	Turn	Script

In the live environment, the FSM is tested for every ground vehicle maneuver twice. As in the simulated environment, the runs at default and improved parameters occur pairwise. The test matrix is in Table 10.

Table 10: Live Environment Test Matrix

Run	Maneuver	Parameters
1	U-turn U-turn	Default Script
2	Straight Straight	Default Script
3	Turn Turn	Default Script
4	Turn Turn	Default Script
5	Straight Straight	Default Script
6	U-turn U-turn	Default Script

3.10 Summary

This chapter outlines the methodology used to answer the research questions. The equipment and materials used to facilitate experimentation are described. The first experiment is a 12-run Plackett-Burman conducted in the simulated environment for the purpose of ensuring the factor levels are set correctly. Second, factors are screened and models are constructed by analyzing data from NCD or DSD experiments conducted in the simulated environment. The detection of quadratic curvature in the first NCD experiment forces a switch to the DSD, which can estimate quadratic effects. Third, the use of simulation software to develop models is validated by conducting near-identical experiments in the live and simulated environments. The run traces from each environment are visually compared to determine parity. Fourth, the FSM is created from

analytical results and the FSM is coded using Python language scripts. Lastly, the FSM is validated by conducting experiments in the both the live and simulated environments. Experimental runs at default and improved parameter settings occur pair-wise. The variances of each pair-wise run are compared using an F-test.

IV. Analysis and Results

4.1 Chapter Overview

This chapter presents the results and analysis of the experiments described in Chapter 3.

4.2 Screening and Model Selection

The 17-run DSD was the design of choice to conduct ten experiments for each combination of ground maneuver and wind direction. Initially, it was believed that a crosswind from the left or right would result in a symmetrical effect on the aircraft. This likely is true for the straight and u-turn maneuvers. However, the turn maneuver was reconsidered. Following the execution of a 90 degree left turn with a crosswind from the left of the aircraft, the crosswind became a headwind. A crosswind from the right resulted in a post-turn tailwind. Thus, the wind effect was not likely symmetrical, so one additional combination was added for the turn maneuver—one that considered a right crosswind and another that considered a left crosswind. Table 11 includes all of the tested combinations.

Table 11: DSD Combinations

Combo	Ground manuever	Wind direction
1	Straight	Headwind
2	Turn	Crosswind right
3	U-turn	Tailwind
4	Straight	Crosswind
5	Turn	Tailwind
6	U-turn	Headwind
7	Straight	Tailwind
8	Turn	Headwind
9	U-turn	Crosswind
10	Turn	Crosswind left

Following the completion of each experiment, a model was selected using the stepwise regression tool in JMP® 11. During model selection, main effects, two-factor interactions, and quadratic effects were considered. In general, maximum bank angle, waypoint radius, waypoint loiter radius, and wind speed were found to be the significant factors that impacted following distance variability, though waypoint radius appeared in only three of the ten models. Not all models contained interaction and quadratic effects. Table 12 shows the complete results. A grayed-out cell indicates the main effect was not significant and not included in the model. Whereas, a factor with "No" or "Yes" entered in both or either the interaction and quadratic cells indicates the main effect was significant and included in the model. In the combination column, the first letter refers to the ground vehicle maneuver (i.e. "S" represents straight), while the second letter(s) refers to the wind direction (i.e. "H" represents headwind).

Table 12: Model Selection Results

Combo	Max bank angle (MBA)		Waypoint radius (WR)		Loiter radius (LR)		Wind speed (WS)	
	Interaction?	Quadratic?	Interaction?	Quadratic?	Interaction?	Quadratic?	Interaction?	Quadratic?
SH	No	No					No	No
ST	No	No					No	No
SC	Yes, WS	Yes	No	No	No	Yes	Yes, MBA	No
TH	No	Yes			No	Yes	No	No
TT	No	No	Yes, WS	No	No	No	Yes, WR	No
TCr	No	Yes			No	No		
TCl	Yes, WS, LR	Yes			Yes, MBA	No	Yes, MBA	No
UH	Yes, WR	No	Yes, MBA	No	Yes, WS	No	Yes, LR	No
UT					Yes, WS	No	Yes, LR	No
UC	Yes, WS	Yes			No	No	Yes, MBA	No

In general, regardless of maneuver, a crosswind resulted in a significant quadratic maximum bank angle effect and an interaction with wind speed. This is intuitive as a crosswind pushes the aircraft laterally off course which the aircraft subsequently corrects by turning (which involves banking) into the wind. A higher wind speed results in requiring a larger maximum bank angle to improve performance. The quadratic maximum bank angle effect results from decreased ability to turn as the bank angle exceeds a certain threshold. This threshold likely exists where the ailerons have more impact on turn ability than the rudder. The straight maneuver resulted in fewest significant factors. This, too, is an intuitive result. It is a less complicated scenario. The aircraft must only compensate for wind and not changes in vehicle direction.

As an example, one of the more interesting cases was the turn maneuver with a left crosswind. Figure 8 shows the selected model and the results of the prediction profiler. To find the settings that resulted in minimizing variance, a desirability function was employed. As each state of the FSM corresponded to a ground vehicle maneuver, wind direction, and wind speed, wind speed was held constant at the low, center, and

high levels. The desirability function then provided the other settings to minimize the following distance variance at the selected wind speed.

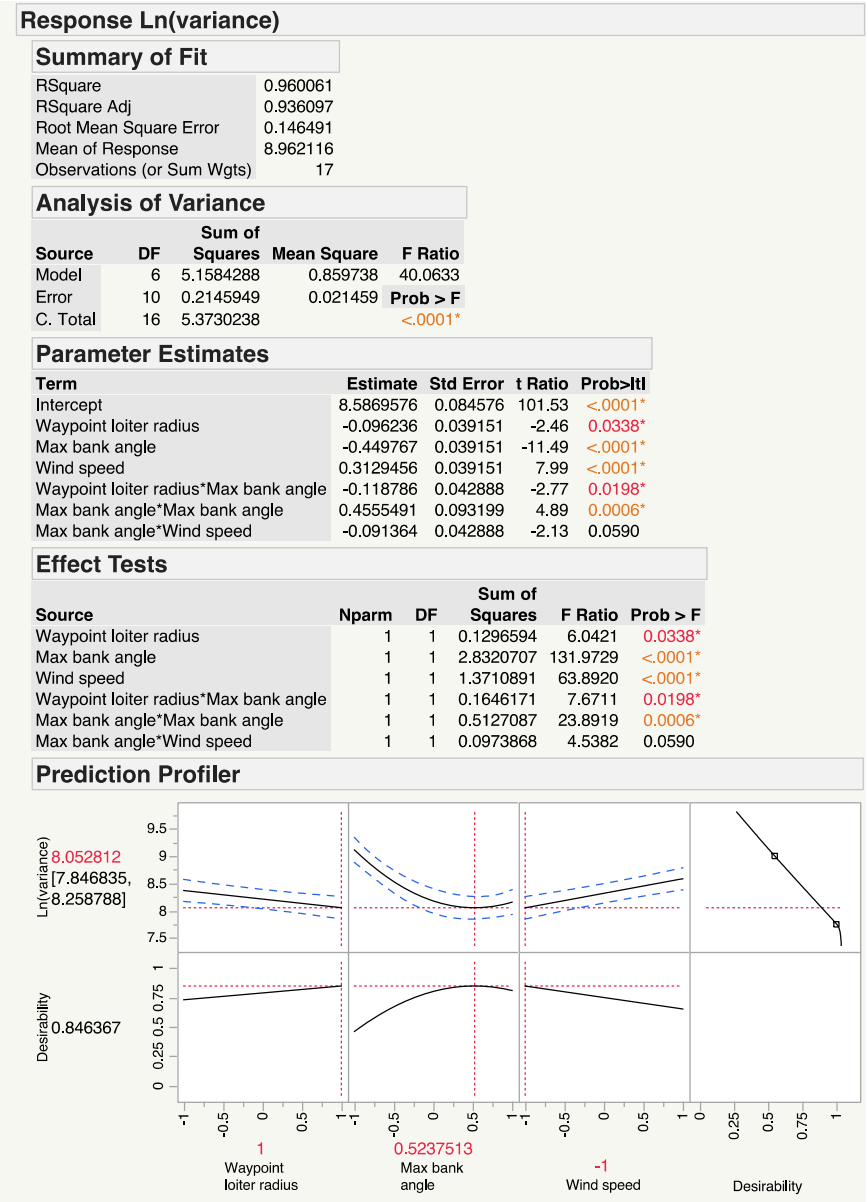


Figure 8: DSD JMP® Output – Turn with Crosswind Left

The process was repeated for each combination, producing Table 13, which represents the states of the FSM. Factors that do not appear in the individual table cells were set to the default value.

Table 13: States Table

Maneuver: Straight

	Wind Direction			
Wind Speed	**Headwind**	**Tailwind**	**Crosswind Right**	**Crosswind Left**
3 knots	Max bank angle=80	Max bank angle=80	Waypoint radius=40 Waypoint loiter radius=30 Max bank angle=70	Waypoint radius=40 Waypoint loiter radius=30 Max bank angle=70
7 knots	Max bank angle=80	Max bank angle=80	Waypoint radius=40 Waypoint loiter radius=30 Max bank angle=55	Waypoint radius=40 Waypoint loiter radius=30 Max bank angle=55
11 knots	Max bank angle=80	Max bank angle=80	Waypoint radius=40 Waypoint loiter radius=30 Max bank angle=40	Waypoint radius=40 Waypoint loiter radius=30 Max bank angle=40

Maneuver: Turn

	Wind Direction			
Wind Speed	**Headwind**	**Tailwind**	**Crosswind Right**	**Crosswind Left**
3 knots	Waypoint loiter radius=70 Max bank angle=65	Waypoint radius=40 Waypoint loiter radius=90 Max bank angle=80	Waypoint loiter radius=90 Max bank angle=70	Waypoint loiter radius=90 Max bank angle=70
7 knots	Waypoint loiter radius=70 Max bank angle=65	Waypoint radius=140 Waypoint loiter radius=90 Max bank angle=80	Waypoint loiter radius=90 Max bank angle=70	Waypoint loiter radius=90 Max bank angle=72
11 knots	Waypoint loiter radius=70 Max bank angle=65	Waypoint radius=140 Waypoint loiter radius=90 Max bank angle=80	Waypoint loiter radius=90 Max bank angle=70	Waypoint loiter radius=90 Max bank angle=75

Maneuver: U-turn

	Wind Direction			
Wind Speed	**Headwind**	**Tailwind**	**Crosswind Right**	**Crosswind Left**
3 knots	Waypoint radius=40 Waypoint loiter radius=90 Max bank angle=40	Waypoint loiter radius=90	Waypoint loiter radius=90 Max bank angle=60	Waypoint loiter radius=90 Max bank angle=60
7 knots	Waypoint radius=40 Waypoint loiter radius=90 Max bank angle=40	Waypoint loiter radius=90	Waypoint loiter radius=90 Max bank angle=70	Waypoint loiter radius=90 Max bank angle=70
11 knots	Waypoint radius=40 Waypoint loiter radius=90 Max bank angle=40	Waypoint loiter radius=30	Waypoint loiter radius=90 Max bank angle=80	Waypoint loiter radius=90 Max bank angle=80

4.3 Simulator Validation

Development of the states in the FSM was accomplished using HIL simulation. The primary purpose of the FSM is to reduce following distance variance in a live environment. As described in the previous chapter, an 8-run experiment was conducted in a live environment and then repeated in the simulated environment. Then, the aircraft GPS tracks from corresponding runs were visually compared.

The conditions on the day of the experiment in the live environment, Camp Atterbury, Indiana, were within the factor settings used to develop models. Specifically, the average wind speed was eight knots. The average wind direction was -60 degrees. The aircraft was given a target altitude of 150 meters and a target airspeed of 18 meters per second. These settings were duplicated in the simulated environment. Taking into account that average wind direction and speed were used in the simulated environment, the results were encouraging.

Figure 9: GPS Tracks – Turn, Run 2

Figure 10: GPS Tracks – U-turn, Run 3

As seen in Figures 9 and 10, the flight paths closely resemble one another. The actual aircraft appears to be able to perform tighter turns than the virtual aircraft. One other difference that is apparent in Figure 11 is that the actual aircraft occasionally turns clockwise. This must be a function of whether or not the aircraft treats the ground vehicle as a regular waypoint or a loiter waypoint. As long as the loiter radius is set to a positive value as it is in this case, the aircraft always loiters counterclockwise as viewed from above. In the simulator, the virtual aircraft seems to always treat the ground vehicle as a loiter waypoint. Whereas, the actual aircraft occasionally treats the ground vehicle as a regular waypoint, allowing it the freedom to turn clockwise or counterclockwise. While these differences are apparent, they do not detract drastically from the notion that models, though built using simulation data, demonstrate the desired impact on following distance variance in the real world. This became apparent when validating the FSM.

Figure 11: GPS Tracks – U-turn, Run 1

4.4 Finite State Machine

The aircraft, as equipped, had no ability to recognize ground vehicle movement, thus, the FSM was limited to autonomously selecting a state using wind direction and speed as inputs. The ground vehicle maneuver information was provided to the aircraft by writing a separate Python script for each maneuver. Foreknowledge of the ground maneuver allowed execution of the appropriate script. Each script contained code written to temporarily store wind direction, wind speed, and waypoint heading from the aircraft telemetry, calculate a wind direction relative to the waypoint heading, and select a set of control parameters based on the relative wind direction and wind speed. The control parameters were maximum bank angle, waypoint radius, and waypoint loiter radius. The control parameter values were those from the states matrix. Any control parameter not listed for a particular state was returned to its default value. The script was executed according to the initial conditions set forth in Chapter 3.

To calculate the relative wind direction, the script was written to transform the wind direction and relative wind direction to the format found in Figure 12. This was done to match the format of the waypoint heading data and facilitate calculations.

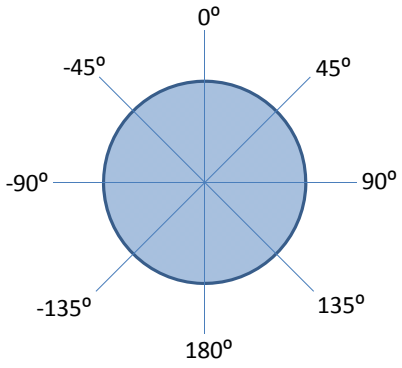


Figure 12: Wind Direction

Python scripts can be found in Appendix A.

4.5 Finite State Machine Validation

The data produced from the aircraft following a ground vehicle comprised an irregular times series with constant mean and variance. A plot of the following distance of one run is found in Figure 13. Sampling the data set at intervals of two to ten seconds resulted in approximately the same sample mean and sample variance. This was done in an attempt to reduce autocorrelation to ensure the sample variance was a meaningful statistic that could be used to perform a hypothesis test.

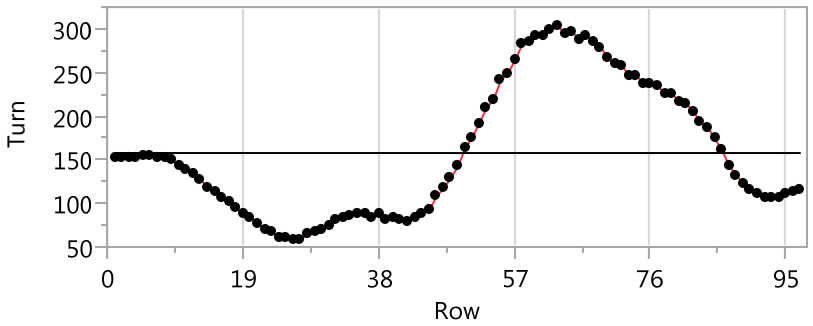


Figure 13: Following Distance Plot – Turn

The presence of constant variance and mean seemed to be intuitive. The aircraft responds to the ground vehicle as it would a regular waypoint. The aircraft moves toward the waypoint and, if it is the last waypoint, the aircraft loiters around the waypoint. In this scenario, the waypoint is a ground vehicle in motion. The aircraft follows the ground vehicle, attempting to close the distance between the two. If the aircraft overtakes the ground vehicle, it circles around and endeavors to loiter at a set radius. However, the ground vehicle is in motion, so the aircraft is unable to loiter and the process continues as the aircraft once again attempts to close the distance from the ground vehicle. This process continues as long as the ground vehicle speed does not exceed the capability of the aircraft.

The entire data set for each run was used to calculate sample variance for use in these hypothesis tests. There was no need to sample from the data set to reduce autocorrelation. A larger sample size improved the power of the F-test and allowed the detection of smaller significant differences in the variance of runs using default control parameters versus runs using modified control parameters.

Two experiments, a 48-run experiment in the simulated environment and a 12-run experiment, were conducted in the live environment. The experiment in the simulated environment produced the results found in Table 14.

Table 14: Script Validation Results – Simulated Environment

Run	Wind Direction	Wind Speed	Maneuver	Parameters	Variance	Decrease?	Significant?
1	Crosswind Left	11	Turn	Default	8043.192	Yes	Yes
	Crosswind Left	11	Turn	Script	4233.793		
2	Headwind	3	Straight	Default	17001.78	Yes	Yes
	Headwind	3	Straight	Script	11137.68		
3	Crosswind Right	11	U-turn	Default	2708.796	Yes	Yes
	Crosswind Right	11	U-turn	Script	1468.434		
4	Crosswind Left	11	Straight	Default	13578.05	Yes	Yes
	Crosswind Left	11	Straight	Script	7812.023		
5	Crosswind Left	11	U-turn	Default	4528.924	Yes	Yes
	Crosswind Left	11	U-turn	Script	1945.296		
6	Headwind	11	Straight	Default	3137.678	No	
	Headwind	11	Straight	Script	5858.507		
7	Headwind	11	Turn	Default	10700.08	Yes	Yes
	Headwind	11	Turn	Script	6824.058		
8	Crosswind Left	3	U-turn	Default	5616.386	No	
	Crosswind Left	3	U-turn	Script	6060.369		
9	Crosswind Left	3	Straight	Default	10337.49	Yes	Yes
	Crosswind Left	3	Straight	Script	6046.178		
10	Crosswind Left	3	Turn	Default	4371.786	Yes	Yes
	Crosswind Left	3	Turn	Script	2035.302		
11	Crosswind Right	3	Straight	Default	15250.33	Yes	Yes*
	Crosswind Right	3	Straight	Script	10432.7		
12	Headwind	11	U-turn	Default	1496.441	Yes	No
	Headwind	11	U-turn	Script	1152.455		
13	Headwind	3	U-turn	Default	2536.41	No	
	Headwind	3	U-turn	Script	3178.739		
14	Crosswind Right	3	Turn	Default	4171.358	Yes	Yes
	Crosswind Right	3	Turn	Script	1929.27		
15	Tailwind	11	U-turn	Default	10832.98	Yes	No
	Tailwind	11	U-turn	Script	10487.79		
16	Tailwind	3	U-turn	Default	5761.988	No	
	Tailwind	3	U-turn	Script	6221.181		
17	Crosswind Right	3	U-turn	Default	4623.165	Yes	No
	Crosswind Right	3	U-turn	Script	4072.253		
18	Tailwind	3	Turn	Default	2890.116	No	
	Tailwind	3	Turn	Script	3050.905		
19	Tailwind	3	Straight	Default	8588.021	No	
	Tailwind	3	Straight	Script	8967.192		
20	Crosswind Right	11	Turn	Default	5483.386	Yes	Yes
	Crosswind Right	11	Turn	Script	2482.297		
21	Tailwind	11	Turn	Default	1626.21	Yes	No
	Tailwind	11	Turn	Script	1395.997		
22	Crosswind Right	11	Straight	Default	14379.76	Yes	No
	Crosswind Right	11	Straight	Script	11758.35		
23	Tailwind	11	Straight	Default	4413.529	Yes	No
	Tailwind	11	Straight	Script	3691.494		
24	Headwind	3	Turn	Default	5584.921	Yes	Yes*
	Headwind	3	Turn	Script	3903.075		

Of the 24 pairwise runs, 18 resulted in reducing the following distance variance. Twelve of the 18 reductions were significant. Runs 11 and 24 were significant with a p-

value of 0.0512 and 0.0615 respectively, while the remaining runs were significant at an alpha level of 0.05. In general, the failure to reduce or significantly reduce variability appeared to be tied to wind direction. Of the 12 failures to significantly reduce the variance, nine occurred under a headwind or tailwind. As previously mentioned, the crosswind was the more interesting case and model selection for this case was straightforward. Also, ground maneuver may have been a factor. Six of the failures to significantly reduce variance occurred during a u-turn maneuver and three additional failures to significantly reduce variance occurred during a straight maneuver. Model selection for these maneuvers was relatively difficult. There seemed to be a tendency to fit noise. The parameter estimates were generally smaller than the turn model parameter estimates, indicating a smaller impact on reducing variability.

The experiment conducted at Camp Atterbury, Indiana produced the results found in Table 15.

Table 15: Script Validation Results – Live Environment

Run	Maneuver	Parameters	Variance	Decrease?	Significant?
1	U-turn U-turn	Default Script	975.394 443.509	Yes	Yes
2	Straight Straight	Default Script	1501.853 1040.129	Yes	Yes*
3	Turn Turn	Default Script	3686.457 1778.945	Yes	Yes
4	Turn Turn	Default Script	2097.117 1379.102	Yes	Yes
5	Straight Straight	Default Script	4109.605 1785.216	Yes	Yes
6	U-turn U-turn	Default Script	2859.928 829.513	Yes	Yes

The live environment results were much more encouraging. All pairwise runs resulted in a significant decrease in following distance variance. Run two was significant with a p-value of 0.0569, while the remaining runs were significant at an alpha level of

0.05. As the FSM was developed for use in a live environment, the fact that it performed better in a live environment is positive. This phenomenon can likely be explained by the earlier observation that the actual aircraft appeared more responsive than the simulated aircraft. The actual aircraft could perform tighter turns. This indicated the actual aircraft likely responded better to changes in the control parameters than the simulated aircraft. Improved response results in further reduced variability.

4.6 Summary

This chapter covered the results of experiments described in the previous chapter. The Plackett-Burman experiment resulted in reducing the low setting for maximum bank angle and verifying proper functioning of the HIL simulation software and hardware. The NCD experiment resulted in an additional reduction in the low setting for maximum bank angle and finding significant curvature. The DSD became the design of choice for model selection. Improved control parameters were selected based on a desirability function applied to the selected models. The improved parameters were organized into a matrix to represent the states of the FSM. The simulator validation experiment showed the simulator sufficiently resembled the live environment, though the actual aircraft was more responsive than the simulated aircraft. The FSM implemented through Python scripts resulted in significantly reducing following distance variability in the live environment when compared to default control parameter settings.

V. Conclusions and Recommendations

5.1 Conclusions of Research

The objective of this research was to improve SUAS performance by enhancing the aircraft's ability to maintain a consistent distance from the ground vehicle in the presence of wind and while the vehicle performed a maneuver. Reducing the variability in following distance through the implementation of an FSM to enhance aircraft autonomy accomplished the objective. Results from sequential designed experiments served as the basis of the FSM.

The experimentation process began with throttle slew rate, waypoint radius, waypoint loiter radius, maximum bank angle, target airspeed, roll time constant, wind speed, and wind direction as factors. In the end, maximum bank angle, waypoint radius, and waypoint loiter radius with settings unique to combinations of wind direction, wind speed, and ground maneuver were determined to be the factors significant to reducing following distance variability.

Each combination of wind direction, wind speed, and ground maneuver formed a state for a total of 36 states. A set of values for maximum bank angle, waypoint radius, and/or waypoint loiter radius characterize each state. Python scripts were written to implement the FSM and allow the SUAS to change the aforementioned control parameters to values best suited for prevailing winds and ground vehicle maneuver.

Based on the results of the live environment validation experiment, the FSM reduced following distance variance by an average of 50 percent when compared to the

SUAS using default parameter settings. This final result demonstrated the efficacy of implementing an FSM to enhance SUAS autonomy.

Additionally, this research effort served as a proof of concept for using a sequential experimentation methodology to develop an FSM for SUAS application. The majority of experiments took place in a simulated environment. This demonstrated that a properly calibrated simulator can serve as a proxy to experimentation in a live environment. Moreover, the experiments conducted in the simulated environment required significantly less time and other resources.

5.2 Recommendations for Future Research

Because the existing APM:Plane firmware did not have the ability to recognize ground vehicle maneuvers, the FSM was limited to autonomously selecting a state using wind direction and speed as inputs. The SUAS operator had to provide the ground vehicle maneuver information to the SUAS by running the Python script unique to the maneuver. A future research effort could improve the scripts to allow the APM autopilot to recognize the ground vehicle maneuver and use that information as a FSM input.

Mission Planner, the ground station software, uses a waypoint to represent the ground vehicle. Waypoint location data is logged. It might be valuable to use the location data, specifically the two most recent data points, to calculate an azimuth. As the ground vehicle changes direction, the azimuth value changes (see Figure 14). The magnitude of change in azimuth could represent a vehicle maneuver. Thresholds for equating azimuth change with a vehicle maneuver could be selected through experimentation. Crossing a threshold would trigger a change in state.

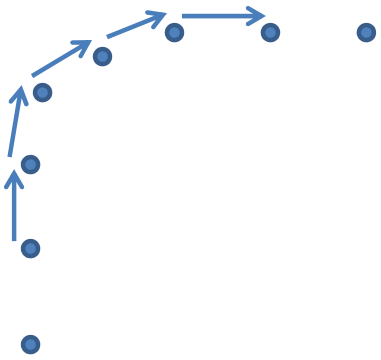


Figure 14: Waypoints and Azimuths

Incorporating the ability to detect ground vehicle maneuvers would greatly enhance the FSM. The Python scripts could be combined into one and run in a loop, constantly monitoring ground maneuver, wind direction, and wind speed. This would further enhance SUAS autonomy and allow use of the FSM to reduce following distance variability over a route made up of multiple ground maneuvers.

Appendix A: Python Scripts

Straight

```
print 'Start Script'

windVel = cs.wind_vel

if cs.wind_dir > 180:
   windDir = cs.wind_dir - 360
else:
   windDir = cs.wind_dir

relWindDir = windDir - cs.target_bearing

if relWindDir < -180:
   relWindDir = 360 + relWindDir
elif relWindDir > 180:
   relWindDir = relWindDir - 360

if relWindDir > -45 and relWindDir < 45: # headwind
   print 'Headwind'
   Script.ChangeParam('LIM_ROLL_CD',8000)
   Script.ChangeParam('WP_RADIUS',90)
   Script.ChangeParam('WP_LOITER_RAD',60)
elif relWindDir >= 45 and relWindDir <= 135: # crosswind right
   print 'Crosswind Right'
   if windVel <= 3:
      Script.ChangeParam('LIM_ROLL_CD',7000)
      Script.ChangeParam('WP_RADIUS',40)
      Script.ChangeParam('WP_LOITER_RAD',30)
   elif windVel > 3 and windVel < 11:
      Script.ChangeParam('LIM_ROLL_CD',5500)
      Script.ChangeParam('WP_RADIUS',40)
      Script.ChangeParam('WP_LOITER_RAD',30)
   elif windVel >= 11:
      Script.ChangeParam('LIM_ROLL_CD',4000)
      Script.ChangeParam('WP_RADIUS',40)
      Script.ChangeParam('WP_LOITER_RAD',30)
elif relWindDir > 135 or relWindDir < -135: # tailwind
   print 'Tailwind'
   Script.ChangeParam('LIM_ROLL_CD',8000)
   Script.ChangeParam('WP_RADIUS',90)
   Script.ChangeParam('WP_LOITER_RAD',60)
```

```
elif relWindDir >= -135 and relWindDir <= -45: # crosswind left
    print 'Crosswind Left'
    if windVel <= 3:
        Script.ChangeParam('LIM_ROLL_CD',7000)
        Script.ChangeParam('WP_RADIUS',40)
        Script.ChangeParam('WP_LOITER_RAD',30)
    elif windVel > 3 and windVel < 11:
        Script.ChangeParam('LIM_ROLL_CD',5500)
        Script.ChangeParam('WP_RADIUS',40)
        Script.ChangeParam('WP_LOITER_RAD',30)
    elif windVel >= 11:
        Script.ChangeParam('LIM_ROLL_CD',4000)
        Script.ChangeParam('WP_RADIUS',40)
        Script.ChangeParam('WP_LOITER_RAD',30)

print 'End Script'
```

Turn

```
print 'Start Script'

windVel = cs.wind_vel

if cs.wind_dir > 180:
    windDir = cs.wind_dir - 360
else:
    windDir = cs.wind_dir

relWindDir = windDir - cs.target_bearing

if relWindDir < -180:
    relWindDir = 360 + relWindDir
elif relWindDir > 180:
    relWindDir = relWindDir - 360

if relWindDir > -45 and relWindDir < 45: # headwind
    print 'Headwind'
    Script.ChangeParam('LIM_ROLL_CD',6500)
    Script.ChangeParam('WP_RADIUS',90)
    Script.ChangeParam('WP_LOITER_RAD',70)
elif relWindDir >= 45 and relWindDir <= 135: # crosswind right
    print 'Crosswind Right'
    Script.ChangeParam('LIM_ROLL_CD',7000)
    Script.ChangeParam('WP_RADIUS',90)
    Script.ChangeParam('WP_LOITER_RAD',90)
```

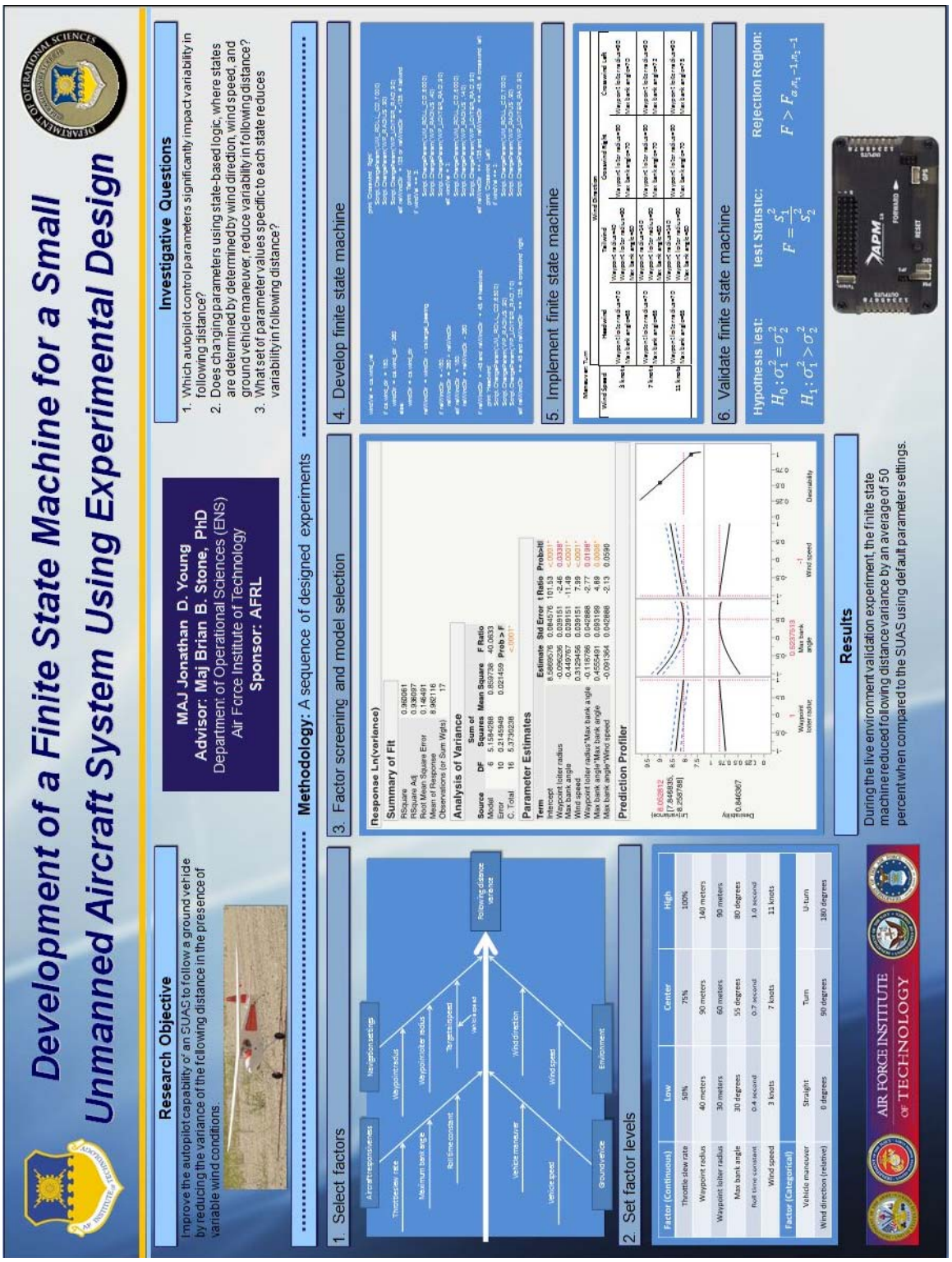
```
elif relWindDir > 135 or relWindDir < -135: # tailwind
    print 'Tailwind'
    if windVel <= 3:
        Script.ChangeParam('LIM_ROLL_CD',8000)
        Script.ChangeParam('WP_RADIUS',40)
        Script.ChangeParam('WP_LOITER_RAD',90)
    elif windVel > 3:
        Script.ChangeParam('LIM_ROLL_CD',8000)
        Script.ChangeParam('WP_RADIUS',140)
        Script.ChangeParam('WP_LOITER_RAD',90)
elif relWindDir >= -135 and relWindDir <= -45: # crosswind left
    print 'Crosswind Left'
    if windVel <= 3:
        Script.ChangeParam('LIM_ROLL_CD',7000)
        Script.ChangeParam('WP_RADIUS',90)
        Script.ChangeParam('WP_LOITER_RAD',90)
    elif windVel > 3 and windVel < 11:
        Script.ChangeParam('LIM_ROLL_CD',7200)
        Script.ChangeParam('WP_RADIUS',90)
        Script.ChangeParam('WP_LOITER_RAD',90)
    elif windVel >= 11:
        Script.ChangeParam('LIM_ROLL_CD',7500)
        Script.ChangeParam('WP_RADIUS',90)
        Script.ChangeParam('WP_LOITER_RAD',90)

print 'End Script'
```

U-turn

```
print 'Start Script'

windVel = cs.wind_vel

if cs.wind_dir > 180:
    windDir = cs.wind_dir - 360
else:
    windDir = cs.wind_dir

relWindDir = windDir - cs.target_bearing

if relWindDir < -180:
    relWindDir = 360 + relWindDir
elif relWindDir > 180:
    relWindDir = relWindDir - 360
```

```
if relWindDir > -45 and relWindDir < 45: # headwind
    print 'Headwind'
    Script.ChangeParam('LIM_ROLL_CD',4000)
    Script.ChangeParam('WP_RADIUS',40)
    Script.ChangeParam('WP_LOITER_RAD',90)
elif relWindDir >= 45 and relWindDir <= 135: # crosswind right
    print 'Crosswind Right'
    if windVel <= 3:
        Script.ChangeParam('LIM_ROLL_CD',6000)
        Script.ChangeParam('WP_RADIUS',90)
        Script.ChangeParam('WP_LOITER_RAD',90)
    elif windVel > 3 and windVel < 11:
        Script.ChangeParam('LIM_ROLL_CD',7000)
        Script.ChangeParam('WP_RADIUS',90)
        Script.ChangeParam('WP_LOITER_RAD',90)
    elif windVel >= 11:
        Script.ChangeParam('LIM_ROLL_CD',8000)
        Script.ChangeParam('WP_RADIUS',90)
        Script.ChangeParam('WP_LOITER_RAD',90)
elif relWindDir > 135 or relWindDir < -135: # tailwind
    print 'Tailwind'
    if windVel < 11:
        Script.ChangeParam('LIM_ROLL_CD',4500)
        Script.ChangeParam('WP_RADIUS',90)
        Script.ChangeParam('WP_LOITER_RAD',90)
    elif windVel >= 11:
        Script.ChangeParam('LIM_ROLL_CD',4500)
        Script.ChangeParam('WP_RADIUS',90)
        Script.ChangeParam('WP_LOITER_RAD',30)
elif relWindDir >= -135 and relWindDir <= -45: # crosswind left
    print 'Crosswind Left'
    if windVel <= 3:
        Script.ChangeParam('LIM_ROLL_CD',6000)
        Script.ChangeParam('WP_RADIUS',90)
        Script.ChangeParam('WP_LOITER_RAD',90)
    elif windVel > 3 and windVel < 11:
        Script.ChangeParam('LIM_ROLL_CD',7000)
        Script.ChangeParam('WP_RADIUS',90)
        Script.ChangeParam('WP_LOITER_RAD',90)
    elif windVel >= 11:
        Script.ChangeParam('LIM_ROLL_CD',8000)
        Script.ChangeParam('WP_RADIUS',90)
        Script.ChangeParam('WP_LOITER_RAD',90)

print 'End Script'
```

Appendix B: Storyboard



Bibliography

[1] D. C. Montgomery, Design and Analysis of Experiments, John Wiley & Sons, Inc., 2012.

[2] B. B. Stone, "No-confounding Designs of 20 and 24 Runs for Screening Experiments and a Design Selection Methodology," Arizona State University, Tempe, 2013.

[3] B. Jones and J. C. Nachtsheim, "A Class of Three-Level Designs for Definitive Screening in the Presence of Second-Order Effects," *Journal of Quality Technology,* vol. 43, no. 1, pp. 1-15, 2011.

[4] D. R. Wright, "CSC216 Course Notes: Finite State Machines," 2005. [Online]. Available: http://www4.ncsu.edu/~drwrigh3/docs/courses/csc216/fsm-notes.pdf. [Accessed 4 February 2015].

[5] 3D Robotics, "APM:Plane Complete Parameter List," [Online]. Available: http://plane.ardupilot.com/wiki/arduplane-parameters/. [Accessed 30 January 2015].

[6] R. Livermore, "Optimal UAV Path Planning for Tracking a Moving Ground Vehicle with a Gimbaled Camera," Air Force Institute of Technology, Wright-Patterson Air Force Base, 2014.

[7] C. J. Neal, "Feasibility of Onboard Processing of Heuristic Path Planning and Navigation Algorithms within SUAS Autopilot Computational Constraints," Air Force Institute of Technology, Wright-Patterson Air Force Base, 2014.

[8] B. D. Lozano, "Improving Unmanned Aircraft Persistence by Enhancing Endurance and Effective Surveillance Using Design of Experiments and Regression Analysis," Air Force Institute of Technology, Wright-Patterson Air Force Base, 2011.

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704–0188

1. REPORT DATE (DD–MM–YYYY)	2. REPORT TYPE	3. DATES COVERED (From — To)
26-03-2015	Master's Thesis	August 2013 – March 2015

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Development of a Finite State Machine for a Small Unmanned Aircraft System Using Experimental Design	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Young, Jonathan D., Major, USA	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/ENS) 2950 Hobson Way WPAFB OH 45433-7765	8. PERFORMING ORGANIZATION REPORT NUMBER AFIT-ENS-MS-15-M-146
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Matthew Clark, Technical Area Lead Verification and Validation of Autonomous Control Systems Autonomous Control Branch Air Force Research Laboratory, AFRL/RQQA 2210 Eighth Street, Room 301 WPAFB OH 45433 Email: matthew.clark.20@us.af.mil	10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RQQA
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Distribution Statement A. Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT
This research presents a methodology for improving the capability of a small unmanned aircraft system (SUAS) to autonomously track a moving ground vehicle. One drawback of the most common open source SUAS autopilot software, APM:Plane, is the inability to maintain a consistent following distance from the target vehicle under varying conditions defined by wind direction, wind speed, and target vehicle maneuver. Finite state machine (FSM) logic was developed to improve the APM:Plane software by reducing the variability in the following distance between the SUAS and the target vehicle. The FSM consists of 36 individual states defined by a combination of four wind directions, three wind speeds, and three ground maneuvers. Once the SUAS enters a particular state, the FSM modifies the default APM:Plane firmware parameter settings to optimal settings. The parameter settings for each state were determined from the statistical analysis of a sequence of designed experiments conducted in a simulated environment. During a real-world software validation experiment, the FSM reduced following distance variance by an average of 50 percent when compared to the default software settings.

15. SUBJECT TERMS
Finite State Machine, Experimental Design, APM Autopilot, SUAS, Autonomous Vehicle Tracking

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	UU	68	Maj Brian B. Stone, PhD, AFIT/ENS
U	U	U			19b. TELEPHONE NUMBER (Include Area Code) (937) 785-3636 brian.stone@afit.edu